



GAU 2171
#2
ZM

P/2291-98

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Masayoshi Kobayashi

Date: July 25, 2001

Serial No: 09/770,017

Group Art Unit: 2171

Filed: January 25, 2001

For: DATA STRUCTURE FOR SEARCH

Assistant Commissioner for Patents
Washington, D.C. 20231

RECEIVED

AUG 02 2001

Technology Center 2100

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In accordance with 35 U.S.C. §119, Applicant confirms the prior request for priority under the International Convention and submits herewith the following document in support of the claim:

Certified Japanese Application No.
2000-021782 Filed January 26, 2000

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to Commissioner of Patents and Trademarks, Washington, D.C. 20231 on July 25, 2001:

Respectfully submitted,

Steven I. Weisburd

Name of applicant, assignee or
Registered Representative

Signature
July 25, 2001

Date of Signature

Steven I. Weisburd

Registration No.: 27,409

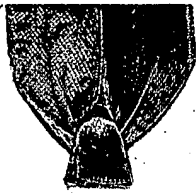
OSTROLENK, FABER, GERB & SOFFEN, LLP

1180 Avenue of the Americas

New York, New York 10036-8403

Telephone: (212) 382-0700

SIW:dr1



日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

09/770017

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 1月26日

出 願 番 号

Application Number:

特願2000-021782

出 願 人

Applicant (s):

日本電気株式会社

RECEIVED

AUG 02 2001

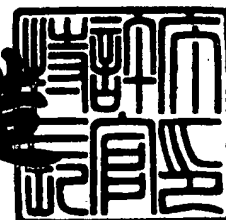
Technology Center 2100

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年10月13日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3084714

【書類名】 特許願

【整理番号】 33509694

【提出日】 平成12年 1月26日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/30

【発明者】

 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

 【氏名】 小林 正好

【特許出願人】

 【識別番号】 000004237

 【氏名又は名称】 日本電気株式会社

【代理人】

 【識別番号】 100088959

 【弁理士】

 【氏名又は名称】 境 廣巳

【手数料の表示】

 【予納台帳番号】 009715

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9002136

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 検索用データ構造構築方法、その装置、機械可読データ記録媒体及び機械可読プログラム記録媒体

【特許請求の範囲】

【請求項 1】 基数検索に用いられるツリー型の検索データ構造の一部に、ツリーの或るノードによって指定されるデータ集合に包含されるような被検索データのすべてを並置したものを並置テーブルとして、前記或るノード以下のすべてのツリー構造の代わりに保持した構造を有する検索用データを記録した機械可読データ記録媒体。

【請求項 2】 複数の被検索データが記録された機械可読データ記録媒体であって、

前記複数の被検索データは、それらの一部が基数検索に用いられるツリー型の検索データ構造で記録され、残りが並置テーブルに記録され、

前記ツリー型の検索データ構造中のノードは、分岐先とする子ノード或いは並置テーブルへのポインタを有し、

前記並置テーブルは、前記複数の被検索データを基数検索に用いるツリー型の検索データ構造とした際の、該検索データ構造中の或るノードによって指定されるデータ集合に包含される全ての被検索データが記録されているものであることを特徴とする機械可読データ記録媒体。

【請求項 3】 請求項 1 または 2 記載の機械可読データ記録媒体において、

前記或るノードは、該或るノードによって指定されるデータ集合に包含される被検索データの数が、該或るノードが位置するレベル以下に存在するレベルの数に、1 ノード当たりの検索時間と前記並置テーブルの 1 エントリ当たりの検索時間との比を乗じた値以下となるノードであることを特徴とする機械可読データ記録媒体。

【請求項 4】 基数検索に用いられるツリー型の検索データ構造の一部に、ツリーの或るノードによって指定されるデータ集合に包含されるような被検索データのすべてを並置したものを並置テーブルとして、前記或るノード以下のすべてのツリー構造の代わりに保持した構造を有する検索用データを構築する検索用

データ構造構築方法であって、

前記或るノードの選択方法として、任意の並置テーブルに含まれる被検索データに対する検索時間中の最大が、並置テーブルに含まれず、ツリーの最下位に位置する被検索データの検索時間の最大を越えないような選択を行うことを特徴とする検索用データ構造構築方法。

【請求項 5】 複数の被検索データに基づいて、基数検索に用いられるツリー型の検索データ構造を構成するツリー構成処理と、

該ツリー構成処理で構成されたツリー型の検索データ構造における最大検索時間を求める最大検索時間算出処理と、

前記ツリー構成処理で構成された前記ツリー型の検索データ構造のノードの中から、それを並置テーブルで置き換えたときの検索データ構造の最大検索時間が、前記最大検索時間算出処理で算出された最大検索時間を越えないノードを選択する置換ノード選択処理と、

前記ツリー構成処理によって構成されたツリー型の検索データ構造中の、前記置換ノード選択処理で選択されたノード以下のツリー構造を、前記選択されたノードによって指定されるデータ集合に包含される全ての被検索データを格納した並置テーブルで置き換える並置テーブル構成処理とを含むことを特徴とする検索用データ構造構築方法。

【請求項 6】 複数の被検索データに基づいて、基数検索に用いられるツリー型の検索データ構造を構成するツリー構成部と、

該ツリー構成部で構成されたツリー型の検索データ構造における最大検索時間を求めると共に、前記ツリー型の検索データ構造のノードの中から、それを並置テーブルで置き換えたときの検索データ構造の最大検索時間が、前記求めた最大検索時間を越えないノードを選択する置換ノード選択部と、

前記ツリー構成部によって構成されたツリー型の検索データ構造中の、前記置換ノード選択部で選択されたノード以下のツリー構造を、前記選択されたノードによって指定されるデータ集合に包含される全ての被検索データを格納した並置テーブルで置き換える並置テーブル構成部とを備えたことを特徴とする検索用データ構造構築装置。

【請求項 7】 コンピュータを、

複数の被検索データに基づいて、基数検索に用いられるツリー型の検索データ構造を構成するツリー構成部、

該ツリー構成部で構成されたツリー型の検索データ構造における最大検索時間を求めると共に、前記ツリー型の検索データ構造のノードの中から、それを並置テーブルに置き換えたときの検索データ構造の最大検索時間が、前記求めた最大検索時間を越えないノードを選択する置換ノード選択部、

前記ツリー構成部によって構成されたツリー型の検索データ構造中の、前記置換ノード選択部で選択されたノード以下のツリー構造を、前記選択されたノードによって指定されるデータ集合に包含される全ての被検索データを格納した並置テーブルで置き換える並置テーブル構成部、

として機能させるためのプログラムを記録した機械可読プログラム記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、検索用データ構造及びその構築技術に関し、特に、最大検索時間を基数検索法と同等以上のものにすることができ、且つ被検索データを格納するためのメモリ使用量を少なくすることができる検索用データ構造及びその構築技術に関する。

【0002】

【従来の技術】

データ検索を行う従来の方法としては、テーブルを利用した検索法や、ツリー型の検索データ構造を利用した基数検索法がある。

【0003】

テーブルを利用した検索法としては、テーブルに被検索データを並べておき、テーブルの先頭エントリから順に検索データと比較していく順次検索法や、テーブルに被検索データをソートして配置しておき、バイナリサーチするバイナリサーチ法がある。しかし、これらのテーブルを利用した検索法は、被検索データの数が多くなると、多大な検索時間を要するという問題がある。

【 0 0 0 4 】

ツリー型の検索データ構造を利用した基数検索法では、検索データを有限個の種類記号からなる記号列とみなし、被検索データの集まりを記号列の先頭の記号から順に記号毎に分類し、分類が進む毎に枝分かれして行くようなツリー状のデータ構造に格納しておく。ツリーのノードには、記号列の種類数に応じた子ノードへのポインタが格納されており、検索データが与えられると、検索データの先頭の記号から順にツリーのノードを辿って探索範囲を狭めて検索を行う。

【 0 0 0 5 】

基数検索法を、最長プレフィクス検索法と呼ばれる検索法を例にとって説明する。最長プレフィクス検索法では、被検索データは、或る固定長（Lビット）以下の任意の長さのビット列で表され、各被検索データは、それぞれ関連情報を持っている。検索データが与えられると、検索データと一致するすべての被検索データの中で最も長いものを求める。今、例えば、図9のような、6つの被検索データ（それぞれはビット列）と関連情報との組がある場合を考える。

【 0 0 0 6 】

従来の基数検索法で、検索データを表す記号の集合を（0，1）にした場合、図9の被検索データは、図10のような2分木で格納される。図の黒丸が、被検索データが格納されているノードを示し、白丸は、分岐を示すノードである。各ノードには、ツリーの子ノードへの2個のポインタと、被検索データを格納しているか否かを示すノード情報フラグが含まれる（格納していればフラグが立つ）。例えば、検索データ“0001111”が与えられると、ノードN1,N2,N3,N4,N6,N7を辿る事になる。途中で、ノード情報フラグが立っているノードを経由すれば、そのノードに格納されている関連情報は記憶しておく。この場合、ノードN3,N6,N7でノード情報フラグが立っていることを検出する（即ち、被検索データと検索データが一致する）が、最も長い（最もツリーの下位にある）ノードN7の関連情報「D」を得る。

【 0 0 0 7 】

検索時間は、ノードを一つ辿る時間をTとすると、レベルNのノードまで検索した場合は、TのN倍である。従って、図10に示すツリー型の検索データ構造

における最大検索時間は、ツリーの最も深いノードがレベル6であるので、6Tとなる。

【0008】

検索において、最大検索時間が重要な性能指標である場合には、レベル数を減らし最大検索時間を減らすために、検索データを一度に1bitより多く調べて行く、N分木 ($N > 2$) が用いられる。

【0009】

例えば、同じ図9の例を、検索データを表す記号を(00,01,10,11)にし、4分木で構成する場合を考える。4分木では、検索データを2bitずつ見てゆくので、奇数ビット長の被検索データは、一度に調べるビット数2の倍数になる長さまで展開されており、展開されたものは、展開前と同じ関連情報をもつ。例えば、図9の被検索データの中の、“00011”はビット長が5であるので、4分木に合うようにビット長6まで展開され、“000110”，“000111”の2つになる。これらは、共に、“00011”の関連情報「D」を持つ。図11は、図9に示した被検索データを4分木で格納するために展開した例を示す。

【0010】

図11に示す被検索データ（展開被検索データ）を4分木で構成した場合の検索データ構造が、図12である。図12における黒丸、白丸の意味は2分木の場合と同様で、各ノードは、そのノードの種類を示すノード情報フラグFGと、4個の子ノードへのポインタと、関連情報とを含んでいる。ノード情報フラグFGは、“1”，“0”の何れかの値をとり、それぞれ、被検索データ格納しているノード、被検索データを格納していないノードを示す。

【0011】

例えば、ノードN2は、値が“1”のノード情報フラグFGと、ノードN3,N4,N5へのポインタ及びヌルポインタ（“11”側への分岐が無いことを示す）からなる4つの子ノードへのポインタと、関連情報Aを持つ。

【0012】

検索においては、たとえば、検索データ“0001111”が与えられると、ノードN1,N2,N4,N7と辿ることにより、ノード情報フラグFGの立っているノードN2,N4,

N7を発見するが、より長い被検索データを示すノードN7の関連情報Dを得る。

【0013】

4分木にした場合には、最大検索時間は、ノードを1つ辿る時間を2分木と同じTとすると、ツリーの最も深いノードがレベル4であるので、4Tとなり、最大検索時間が6Tであった2分木に比較して最大検索時間を2/3に短縮することができる。

【0014】

しかしながら、一般的に、被検索データのツリーが粗な場合、最大検索時間を短くするために、N分木のNを大きく取ると、被検索データをN分木構造に合わせるための展開によって、同じ関連情報をもつノードが増加し、また、1つのノードの大きさも、ポインタ数がNに比例して増大することから、メモリ使用量が増大してしまう。

【0015】

例えば、ノード情報フラグFGが1bit、ポインタが1つ6bit、関連情報が8bitとすると、2分木、4分木の1ノード当たりのビット数は、それぞれ $1 + 6 \times 2 + 8 = 21 \text{ bit}$ 、 $1 + 6 \times 4 + 8 = 33 \text{ bit}$ となり、また、図10、図12に示す2分木、4分木の総ビット数は、ノード数が15ノード、14ノードであるので、 $21 \times 15 = 315 \text{ bit}$ 、 $33 \times 14 = 462 \text{ bit}$ となる。また、被検索データを格納するメモリの1アドレス当たりのビット数が32ビットであるとする、2分木の場合には、15アドレスで済むが、4分木の場合は、28アドレスが必要になってしまう。

【0016】

【発明が解決しようとする課題】

上述したような、従来のN分木のツリーデータ構造では、ツリーのノードに、子ノードへのポインタを格納するため、メモリ使用量が大きくなるという問題があった。また、各ノードそれぞれに必要な子ノードへのポインタの数は、Nに比例して増加する。このため、検索速度を速める目的でNを大きくすれば、1ノード内のポインタの数が増加する上、プレフィクス展開により、多くのノードを使用することになる為、この問題がより顕著になっていた。

【 0 0 1 7 】

【発明の目的】

そこで、本発明の目的は、最大検索時間を基数検索法と同等以上にし、且つ、メモリの使用量を基数検索に比較して少なくすることにある。

【 0 0 1 8 】

【課題を解決するための手段】

本発明の機械可読データ記録媒体は、メモリ使用量を削減するため、

基数検索に用いられるツリー型の検索データ構造の一部に、ツリーの或るノードによって指定されるデータ集合に包含されるような被検索データのすべてを並置したものを並置テーブルとして、前記或るノード以下のすべてのツリー構造の代わりに保持した構造を有する検索用データが記録される。

【 0 0 1 9 】

被検索データが、検索データの取りうるすべての値の中で、粗に分布している場合、従来の基数検索法のための、ノード間をポインタで接続するような複雑なツリー型の検索データ構造よりも、データを並置する、並置テーブルの方が、ツリー構造のような子へのポインタが不要であるため、データ構造の大きさを小さくすることができる。

【 0 0 2 0 】

また、本発明の機械可読データ記録媒体は、最大検索時間を基数検索法と同等以上にし、且つ、メモリの使用量を削減できるようにするため、

前記或るノードは、該或るノードによって指定されるデータ集合に包含される被検索データの数が、該或るノードが位置するレベル以下に存在するレベルの数に、1ノード当たりの検索時間と前記並置テーブルの1エントリ当たりの検索時間との比を乗じた値以下となるという条件を満たしている。

【 0 0 2 1 】

並置テーブルは、検索データの数が十分小さいと、従来のツリー型の検索データ構造を用いる基数検索法より検索時間が短く、大きくなると急激に検索時間が増大する傾向がある。このため、従来のツリー型の検索データ構造で、或るノードより先に存在する被検索データの数が十分小さい時（或るノードより先に存在

する被検索データの数、上記或るノードが位置するレベル以下に存在するレベルの数に、1ノード当たりの検索時間と並置テーブルの1エントリ当たりの検索時間との比を乗じた値以下の時)には、並置テーブルに置き換えることで、最大検索時間を増大させることなく、データ構造を小さくすることができる。

【0022】

【発明の実施の形態】

次に本発明の実施の形態について図面を参照して詳細に説明する。

【0023】

図1は本発明の第1の実施の形態のブロック図であり、コンピュータ11と、キーボード等の入力装置21と、ディスプレイ装置等の出力装置31とを備えている。コンピュータ11は、検索部12と、主記憶装置等の記録媒体13とを備えている。

【0024】

図2は記録媒体13に記録された被検データ構造の一例を示す図であり、被検索データの一部は、基数検索に用いられる4分木のツリー型の検索データ構造で記録され、残りは、並置テーブルN2'に記録されている。尚、本実施の形態では、検索データの最大ビット長 $L = 8 \text{ bit}$ とする。

【0025】

図2の各ノードN1,N8,N9,N10,N11,N12,N13,N14は、ノードの種類を示すノード情報フラグFGと、“00”方向，“01”方向，“10”方向，“11”方向の4個のポインタと、関連情報とを含んでいる。ノード内のノード情報フラグFGは、“0”，“1”の何れかの値をとり、それぞれ、被検索データを格納していないノード、被検索データを格納しているノードを示す。尚、本実施の形態においては、ノードの格納されている情報の先頭2bitがノード情報フラグFGに割り当てられているとする。

【0026】

例えば、ノードN1は、被検索データに対応する関連情報を格納していないノードであるので、値“0”のノード情報フラグFGを持ち、“00”方向，“01”方向，“10”方向，“11”方向のポインタとしてそれぞれ並置テーブルN2'へのポインタ，NULL（無効ポインタ），NULL，ノードN8へのポインタを持つ。更に、被検索

データを格納していないため、関連情報には無効値が入っている。

【 0 0 2 7 】

また、例えば、ノードN8は、被検索データを格納していないノードであるので、値“0”のノード情報フラグFGを持ち、“00”方向，“01”方向，“10”方向，“11”方向へのポインタとしてそれぞれNULL，ノードN9へのポインタ，NULL，ノードN10 へのポインタを持つ。更に、被検索データを格納していないため、関連情報には無効値が入っている。

【 0 0 2 8 】

また、ノードN11 は、被検索データを格納しているノードであるので、値“1”のノード情報フラグFGを持ち、“00”方向，“01”方向，“10”方向，“11”方向へのポインタとしてNULLを持つ。更に、被検索データ110100を格納しているので、関連情報にはEが入っている。

【 0 0 2 9 】

並置テーブルN2' は、被検索データを並置したものである。被検索データの並置方法はさまざま考えられるが、本実施の形態では、連続したアドレスに並置するものとする。並置テーブルN2' の各エントリは、ノード情報フラグFG，最終エントリフラグ，データビット列，検索ビット長，関連情報の5つのフィールドを持つ。尚、本実施の形態においては、エントリの先頭2 bit がノード情報フラグFGに割り当てられているとする。

【 0 0 3 0 】

並置テーブルN2' 内のノード情報フラグFGは、常に値“2”を持つ。最終エントリフラグは、並置テーブルN2' の最終エントリのみ“1”で、それ以外のエントリは“0”の値をとる。データビット列は、被検索データのビット列に、検索データの最大ビット長L（この例では、L=8）までビット“0”を付加したビット列である。検索ビット長は、データビット列中の最上位ビットから何ビットが被検索データであるかを示している。関連情報フィールドは、被検索データに対応する関連情報を格納するフィールドである。

【 0 0 3 1 】

図2に示した検索データ構造は、図12に示したノードN2以下のツリー構造を

並置テーブルN2'で置き換えた点以外は、図12と同じ構造を持っている。並置テーブルN2'には、図12のノードN2が表すデータの集合、即ち、“00”で始まるビット列という集合に包含される被検索データが全て格納されている。

【0032】

次に、検索時の動作を説明する。本実施の形態では、ツリー型の検索データ構造を辿って行く場合（分岐先のノード情報フラグFGの値が“0”或いは“1”の場合）は、従来と同じく、検索データのビットを2bitずつ調べながらツリー構造を辿って行く。しかし、分岐先のノード情報フラグFGの値が“2”である場合には、分岐先が並置テーブルN2'であるので、並置テーブルN2'の各エントリを検索データを使用して単純検索して行く。

【0033】

図3は検索部12の処理例を示す流れ図であり、以下各図を参照して検索時の動作を説明する。

【0034】

入力装置21から検索データが入力されると、検索部12は、図3の流れ図に示すように、記録媒体13に格納されている検索データ構造のルートノードの内容を入力し、ノード情報フラグFGの値が“0”であるか、“1”であるかを調べる（A1、A2）。そして、ノード情報フラグFGの値が“0”であった場合は、直ちにA4の処理を行い、“1”であった場合は、ルートノード中の関連情報と一致ビット長との組を、図示を省略した記憶領域に保存した後（A3）、A4の処理を行う。尚、一致ビット長は、ノードのレベルに応じた値をとるものであり、レベルNLのノードの一致ビット長は、 $2 \times (NL - 1)$ となる。

【0035】

A4では、4つのポインタの内の、検索データの先頭2bitのビットパターンと対応するポインタを取得し、次のA5ではポインタの値がNULLであるか否かを調べる。

【0036】

そして、ポインタの値がNULLの場合（A5がYES）は、保存している関連情報の内の、それと組になって保存されている一致ビット長が最も長いものを検索

結果として出力装置31等へ出力した後（A12）、処理を終了する。これに対して、ポインタの値がNULLでない場合（A5がNO）は、A4で取得したポインタが指し示すノード或いは並置テーブルN2'の先頭エントリの内容を入力し（A6）、ノード情報フラグFGの値が“2”であるか否かを調べる（A7）。尚、A6の実行時点では、A4で取得したポインタのポイント先が、ノードであるのか、並置テーブルN2'の先頭エントリであるのかが不明であるため、ポインタが指し示すアドレスから、1ノード分のデータを格納するのに必要なアドレス数と1エントリ分のデータを格納するのに必要なアドレス数との内の、大きな方のアドレス数分のアドレスをアクセスし、データを入力する。例えば、1ノード分のデータを格納するのに必要なアドレス数が「2」で、1エントリ分のデータを格納するのに必要なアドレス数が「1」である場合は、ポインタが指し示すアドレスXと次のアドレス（X+1）とをアクセスすることになる。

【0037】

そして、ノード情報フラグFGが“2”でなかった場合（A7がNO）は、再びA2の処理を行い、ツリー構造を辿る。これに対してノード情報フラグFGが“2”であった場合（A7がYES）は、検索データと並置テーブルN2'の先頭エントリに格納されているデータビット列との、検索ビット長によって示される部分が一致するか否かを判断する（A8）。

【0038】

A8で不一致と判断した場合は、現在処理対象にしているエントリが最終エントリであるか否かを最終エントリフラグに基づいて判断する（A10）。これに対して、A8で一致と判断した場合は、現在処理対象にしているエントリ中の関連情報と一致ビット長（エントリ中の検索ビット長と同じ値を有する）との組を記憶領域に保存した後（A9）、A10の処理を行う。

【0039】

A10で最終エントリであると判断した場合は、保存している関連情報の内の、それと組になって登録されている一致ビット長が最も長いものを検索結果として出力装置31等へ出力した後（A12）、処理を終了する。これに対して、A10で最終エントリでないと判断した場合は、次のエントリを処理対象にした後（

A 1 1)、再び A 8 の処理を行う。

【0040】

次に、具体例を挙げて検索時の動作を説明する。

【0041】

今、例えば、本発明の記録媒体13に図2に示す検索データ構造が記録されている状態において、検索データとして“00011111”が入力されたとする。

【0042】

検索部12は、検索データ“00011111”が入力されると、ルートノードN1の内容を入力し、ノード状態フラグFGの値を調べる(A 1, A 2)。この例の場合、ノード状態フラグFGの値が“0”であるので、検索部12は、A 4 の処理を行う。A 4 では、検索データの先頭2ビットが“00”であるので、ノードN1に設定されている4個のポインタの内の、“00”方向のポインタを取得する。ノードN1の“00”方向のポインタは、「N 2」であるので、検索部12は、並置テーブルN2' の先頭エントリの内容を入力する(A 6)。

【0043】

A 6 で入力した並置テーブルN2' の先頭エントリのノード情報フラグFGの値は“2”であるので(A 7 が Y E S)、検索部12は、検索データと先頭エントリ中のデータビット列との、検索ビット長によって示される部分が一致するか否かを調べる(A 8)。この例の場合、先頭エントリに格納されているデータビット列、検索ビット長はそれぞれ“00000000”，「2」であり、検索データ“00011111”の上位2 bit “00”と、データビット列“00000000”の上位2 bit “00”とは一致するので、A 8 の判断結果は Y E S となる。

【0044】

A 8 の判断結果が Y E S となると、検索部12は、先頭エントリ中の関連情報 A と一致ビット長「2」との組を記憶領域に保存した後(A 9)、第2番目のエントリを処理対象にする(A 1 1)。以下、最終エントリとなるまで(A 1 0 が Y E S となるまで)、並置テーブルN2' の各エントリを対象にして前述した処理と同様の処理を繰り返し行う。この例の場合、第3番目、第4番目のエントリを処理対象にした時、A 8 の判断結果が Y E S となり、関連情報「C」と一致ビット

長「4」との組、関連情報「D」と一致ビット長「5」との組が記憶領域に保存される。

【0045】

そして、最終エントリまで処理を行うと（A10がYES）、検索部12は、一致ビット長が最も長い関連情報「D」を検索結果として出力装置31等に出力する（A12）。

【0046】

尚、上述した説明では、A3，A9において、関連情報と検索ビット長との組を記憶領域に保存するようにしたが、並置テーブルN2'の各エントリが検索ビット長が短い順にソートされている場合には、関連情報のみを記憶領域に保存し、最も後に保存した関連情報を検索結果として出力するようにしても良い。

【0047】

ところで、検索時間は、メモリアクセス時間が大部分を占めるため、並置テーブルN2'の1エントリを調べる時間は、ツリーのノードを1つ辿る時間Tと同じと考えられる。並置テーブルN2'を調べる場合の最大検索時間は、ノードN1を調べるのにT、並置テーブルN2'を調べるのに4Tかかるため、5Tとなる。並置テーブルN2'を調べない場合は、ノードN11,N12,N13,N14のいずれかに到達するような検索の場合が最大の検索時間となり、その時間は、4Tである。このため、全体の最大検索時間は、5Tとなる。

【0048】

【第1の実施の形態の効果】

本実施の形態の検索データ構造における並置テーブルN2'は、被検索データを並べているだけであり、従来のツリー構造のように、子ノードへのポインタが不要であるため、データ構造の大きさを小さくすることが可能である。

【0049】

例えば、図2に示す本実施の形態の検索データ構造と、図2と同じ被検索データを対象にしている図12に示す従来の検索データ構造とを、データ構造の大きさを比較する。本実施の形態における各ノードは、ノード情報フラグFGが並置テーブルN2'の識別の為に1bit増加して2bitになる以外は、図12に示した従

来のノードと同じ構成を有する。従って、本実施の形態におけるノードのビット数は、従来のノードのビット数が33ビットであるので、34ビットとなる。また、並置テーブルN2'の1エントリ当たりのビット数は、ノード情報フラグFGを2bit、最終エントリフラグを1bit、データビット列を8bit（検索データの最大ビット長L=8としたので）、検索データ長を4bit（0から8を示すため）、関連情報を8bitとすると、23bitとなる。従って、ノード数が8個、エントリ数が4個である、図2に示す本実施の形態の検索データ構造において必要になるビット数は、 $34 \times 8 + 23 \times 4 = 364 \text{ bit}$ となる。これに対して、図12に示した従来例において必要になるビット数は、1ノード当たりのビット数が33bitで、ノード数が14個であるので、 $33 \times 14 = 462 \text{ bit}$ となる。以上の事から、本実施の形態によれば、検索データ構造の大きさを、約22%削減することができる。

【0050】

また、本実施の形態によれば、検索データ構造を格納する際に使用するメモリのアドレス数を削減することもできる。例えば、メモリの1アドレス当たりのビット数が32bitであるとする、図12に示す従来の検索データ構造においては、ノード数が14個で、1ノード分のデータを格納するのに、2アドレスを必要とするので、 $2 \times 14 = 28$ アドレスが必要となる。これに対して、図2に示す本実施の形態の検索データ構造では、ノード数が8個、並置テーブルN2'のエントリ数が4個で、1ノード分のデータ、1エントリ分のデータを格納するのに必要なアドレス数はそれぞれ2個、1個であるので、検索データ構造を格納するために必要となるアドレス数は、 $2 \times 8 + 1 \times 4 = 20$ アドレスとなる。以上の事から、本実施の形態によれば、使用するメモリのアドレス数を約29%削減することができる。

【0051】

次に、本発明の第2の実施の形態を説明する。図4は本発明の第2の実施の形態のブロック図であり、コンピュータ41と、キーボード等の入力装置51と、ディスプレイ装置等の出力装置61と、記録媒体71とから構成されている。

【0052】

コンピュータ41は、データ構築部42と、検索部43と、主記憶装置等の記録媒体44とを備えている。

【 0 0 5 3 】

記録媒体44には、図5に示すような、被検索データと関連情報との組が予め格納されている。

【 0 0 5 4 】

検索部43は、図1に示した検索部12と同様の機能を有している。

【 0 0 5 5 】

データ構築部42は、制御部421 と、ルートポインタレジスタ423 を有するツリー構成部422 と、置換ノード選択部424 と、並置テーブル構成部425 とを備えている。

【 0 0 5 6 】

制御部421 は、データ構築部42内の各部の動作を制御する機能を有する。

【 0 0 5 7 】

ツリー構成部422 は、記録媒体44に格納されている被検索データと関連情報との組に基づいて、基数検索に用いられる4分木のツリー型の検索データ構造を記憶媒体44上に構成する機能を有する。

【 0 0 5 8 】

置換ノード選択部424 は、ツリー構成部422 が構成したツリー型の検索データ構造における最大検索時間を求める機能を有すると共に、上記ツリー型の検索データ構造のノードの中から、それを並置テーブルで置き換えた場合の検索データ構造の最大検索時間が上記求めた最大検索時間を越えず、且つメモリ使用量が低減されるノードを選択する機能を有する。

【 0 0 5 9 】

並置テーブル構成部425 は、ツリー構成部422 が構成したツリー型の検索データ構造中の、置換ノード選択部424 で選択されたノード以下のツリー構造を、上記選択されたノードによって指定されるデータ集合に包含される全ての被検索データを格納した並置テーブルで置き換える機能を有する。

【 0 0 6 0 】

記録媒体71は、ディスク、半導体メモリ、その他の記録媒体であり、コンピュータ41を検索用データ構造構築装置として機能させるためのプログラムが記録されている。このプログラムは、コンピュータ41によって読み取られ、コンピュータ41の動作を制御することで、コンピュータ41上に、制御部421、ツリー構成部422、置換ノード選択部424、並置テーブル構成部425、検索部43を実現する。

【 0 0 6 1 】

図6はデータ構築部42の処理例を示す流れ図であり、以下各図を参照して検索データ構造構築時の動作を説明する。

【 0 0 6 2 】

検索データ構造の構築時、データ構築部42内の制御部421は、ツリー構成部422に対して、ツリー型の基数検索用の検索データ構造を記録媒体44上に構成するように指示する。これにより、ツリー構成部422は、記録媒体44に格納されている、図5に示す被検索データと関連情報との組に基づいて、図7に示すようなツリー型の基数検索用の検索データ構造を記録媒体44上に構成する（図6，B1）。尚、図7に示す各ノードN1～N13が有している情報は、図2に示した各ノードが有している情報と同様のものである。その後、ツリー構成部422は、図7に示す検索データ構造の最上位のノードであるルートノードN1へのポインタを、ルートポインタレジスタ423に格納する（B2）。ツリー構成部422の処理が終了すると、制御部421は、置換ノード選択部424を起動する。

【 0 0 6 3 】

これにより、置換ノード選択部424は、ルートポインタレジスタ423中のポインタによってルートノードN1が指し示されている、図7の検索データ構造の最大検索時間を求める（B3）。この最大検索時間 T_{max} は、ツリーのノードを1つ辿る時間 T_n に、ツリーの最も深いノードのレベル N_{Lmax} を乗じることにより求めることができ、 $T_{max} = T_n \times N_{Lmax}$ となる。

【 0 0 6 4 】

次に、置換ノード選択部424は、図7に示す検索データ構造中に存在する各ノードを予め決められている規則（例えば、上位レベルに存在するノードを優先し、同一レベルに存在するノードについては、“00”方向，“01”方向，“10”方

向, “11” 方向の順にするという規則) に従って順次処理対象にし、下記の条件 a, b を満たすノードを 1 つ探し出し、探し出したノードへのポインタ r を並置テーブル構成部 425 へ渡す (B 4)。

【 0 0 6 5 】

a. 処理対象にしているノードを並置テーブルに置き換えた場合、置き換え後の検索データ構造の最大検索時間が B 3 で求めた最大検索時間 T_{\max} より長くない。

b. 処理対象にしているノードを並置テーブルに置き換えた場合、置き換え後のメモリ使用量の方が、置き換え前のメモリ使用量よりも少なくなる。

【 0 0 6 6 】

尚、処理対象にしているノードが、上記条件 a を満たすか否かの判断は、例えば、次のようにして行う。処理対象としたノードによって指定されるデータ集合に含まれる被検索データの数 (並置テーブルのエントリ数と同数) が、処理対象のノードが位置するレベル以下に存在するレベルの数に、1 ノード当たりの検索時間 T_n と並置テーブルの 1 エントリ当たりの検索時間 T_e との比 $K = T_e / T_n$ を乗じた値以下になった場合、条件 a を満たすと判断する。

【 0 0 6 7 】

制御部 421 は、置換ノード選択部 424 の処理が終了すると、並置テーブル構成部 425 を起動する。

【 0 0 6 8 】

これにより、並置テーブル構成部 425 では、ポインタ r が指し示すノード以下のツリー構造を解放し、上記ノードによって指定されるデータ集合に含まれる被検索データと関連情報との組を記録媒体 44 上から探し、並置テーブルとして再構成する (B 5)。その後、並置テーブル構成部 425 は、B 5 で構成した並置テーブルを、記録媒体 44 上のポインタ r によってポイントされる位置に保存する (B 6)。制御部 421 は、上で示した、置換ノード選択部 424 と並置テーブル構成部 425 の起動を任意回繰り返しても良い。尚、置換ノード選択部 424 は、制御部 421 によって複数回起動された場合は、既に選択しているノードは処理対象から除外して、前述した処理を行う。

【 0 0 6 9 】

図 8 は、ツリー構成部 422 が構築した図 7 の検索データ構造に対して、置換ノード選択部 424 と並置テーブル構成部 425 とを 1 度ずつ動作させた時に得られた検索データ構造の例を示した図である。図 8 に示すように、ノード N6 以下のツリー構造が並置テーブル N6' で置き換えられている。

【 0 0 7 0 】

次に、上述したデータ構築部 42 の動作を具体例を挙げて説明する。尚、以下の説明では、ノード当たりの検索時間 T_n 、1 エントリ当たりの検索時間 T_e は、共に T であるとする。

【 0 0 7 1 】

ツリー構成部 422 が、記録媒体 44 上に、図 7 に示す基数検索用のツリー型の検索データ構造を構築し、ルートポインタレジスタ 423 にルートノード N1 へのポインタを格納すると（図 6 の B 1, B 2）、置換ノード選択部 424 は、図 7 の検索データ構造における最大検索時間 T_{max} を求める（B 3）。図 7 の検索データ構造の最大レベルは「4」であるので、最大検索時間 $T_{max} = 4 T$ となる。

【 0 0 7 2 】

次に、ノード N1 から幅優先探索でノードを調べて行く。ノード N1 を並置テーブルに置き換えると、並置テーブルのエントリ数は「8」となり、置き換え後の検索データ構造における最大検索時間は $8 T$ となる。これは、最大検索時間を増加させることになり、条件 a を満たさないなので、置換ノード選択部 424 は、次のノード N2 を処理対象にする。

【 0 0 7 3 】

ノード N2 を並置テーブルに置き換えるすると、並置テーブルのエントリ数は「4」となる。従って、ノード N2 を並置テーブルに置き換えた時の検索データ構造における最大検索時間は、ノード N1 の検索時間 T と、エントリ数が「4」の並置テーブルの検索時間 $4 T$ との和である $5 T$ となり、B 3 で求めた最大検索時間 $T_{max} = 4 T$ を越えてしまう。このため、このノード N2 も選択せず、幅優先探索を進め、ノード N6 を処理対象ノードにする。

【 0 0 7 4 】

ノードN6を並置テーブルに置き換えた場合の、並置テーブルのエントリ数は「2」となる。従って、並置テーブルの最終エントリまで検索するのに要する時間は、ノードN1の検索時間Tと、エントリ数が「2」の並置テーブルの検索時間2Tとの和である3Tとなる。このことは、ノードN6を並置テーブルで置き換えても、置き換え後の検索データ構造における最大検索時間が、B3で求めた最大検索時間 T_{\max} を越えないことを示している。従って、ノードN6は上記した条件aを満たすノードとなる。

【0075】

更に、ノードN6,N7,N8,N9をなくすことによる削減ビット数は、1ノード当たりのビット数を第1の実施の形態と同じ34bitとすると、 $34 \times 4 = 136$ bitとなる。これに対して、並置テーブルN6'の為に必要なビット数は、並置テーブルの1エントリ当たりのビット数が第1の実施の形態と同じ23bitであるとすると、 $23 \times 2 = 46$ bitとなる。よって、 $136 - 46 = 90$ bitの削減が達成できる。従って、置換ノード選択部424は、ノードN6を選択し、ノードN6へのポインタを並置テーブル構成部425へ渡す。

【0076】

並置ノード構成部425は、ノードN6以下のツリー構造を構成するノードN6,N7,N8,N9のメモリ領域を解放する。そして、ノードN6が示すデータ集合が、“10”で始まる全てのビット列であることから、この集合に含まれる被検索データと関連情報との組を、記録媒体44から探し出す。この例の場合、記録媒体44には、図5に示す被検索データと関連情報との組が記録されているので、並置テーブル構成部425は、被検索データと関連情報との組として(1000,E),(10000,F)を見つける。その後、見つけ出した被検索データと関連情報の組それぞれに対して、ノード情報フラグFG、最終エントリフラグ、データビット列、検索ビット長、関連情報からなる並置テーブルN6'の1エントリ分のデータを作成する。尚、データビット列は、探し出した被検索データに、“0”を付加してその長さを検索データの最大ビット長と一致させたものである。その後、作成した各データを並置テーブルN6'として、記録媒体44上の、ポインタN6によって示される位置に格納する。

【 0 0 7 7 】

【第2の実施の形態の効果】

上に示したとおり、従来の基数検索法を用いたデータ構造の最大検索時間 $4T$ を変えずに、メモリ領域を 90bit 節約している。即ち、最大検索時間を増加させる事無く、メモリ使用量を削減することができる。また、本実施の形態によれば、検索データ構造を格納する際に使用するメモリのアドレス数を削減することもできる。例えば、メモリの1アドレス当たりのビット数が 32bit であるとする、図12に示す従来の検索データ構造においては、ノード数が14個で、1ノード分のデータを格納するのに、2アドレスを必要とするので、 $2 \times 14 = 28$ アドレスが必要となる。これに対して、図8に示す本実施の形態の検索データ構造では、ノード数が9個、並置テーブル $N6'$ のエントリ数が2個で、1ノード分のデータ、1エントリ分のデータを格納するのに必要なアドレス数はそれぞれ2個、1個であるので、検索データ構造を格納するために必要となるアドレス数は、 $2 \times 9 + 1 \times 2 = 20$ アドレスとなる。以上の事から、本実施の形態によれば、使用するメモリのアドレス数を約 29% 削減することができる。

【 0 0 7 8 】

【発明の効果】

本発明の第1の効果は、検索データ構造を格納するためのメモリ使用量を削減できるということである。その理由は、基数検索に用いられるツリー型の検索データ構造の一部を、子ノードへのポインタが不要な、データサイズを小さくすることができる並置テーブルで置き換えているからである。

【 0 0 7 9 】

本発明の第2の効果は、最大検索時間を基数検索法と同等以上にし、且つ、メモリの使用量を削減できるということである。その理由は、ツリー型の検索データ構造で、或るノードより先に存在する被検索データの数が十分小さい時（或るノードより先に存在する被検索データの数が、上記或るノードが位置するレベル以下に存在するレベルの数に、1ノード当たりの検索時間と並置テーブルの1エントリ当たりの検索時間との比を乗じた値以下の時）には、並置テーブルに置き換えるようにしているからである。

【図面の簡単な説明】

【図 1】

本発明の第 1 の実施の形態のブロック図である。

【図 2】

記録媒体 13 に記録された検索データ構造の一例を示す図である。

【図 3】

検索部 12 の処理例を示す流れ図である。

【図 4】

本発明の第 2 の実施の形態のブロック図である。

【図 5】

被検索データと関連情報との組の一例を示す図である。

【図 6】

データ構築部 42 の処理例を示す流れ図である。

【図 7】

ツリー構成部 423 によって構成される検索データ構造の一例を示した図である。

【図 8】

記録媒体 44 に記録された検索データ構造を示す図である。

【図 9】

被検索データと関連情報との組の一例を示す図である。

【図 1 0】

従来の 2 分木の検索データ構造の一例を示す図である。

【図 1 1】

図 9 に示す被検索データと関連情報との組を 4 分木で格納できるように展開した図である。

【図 1 2】

従来の 4 分木の検索データ構造の一例を示す図である。

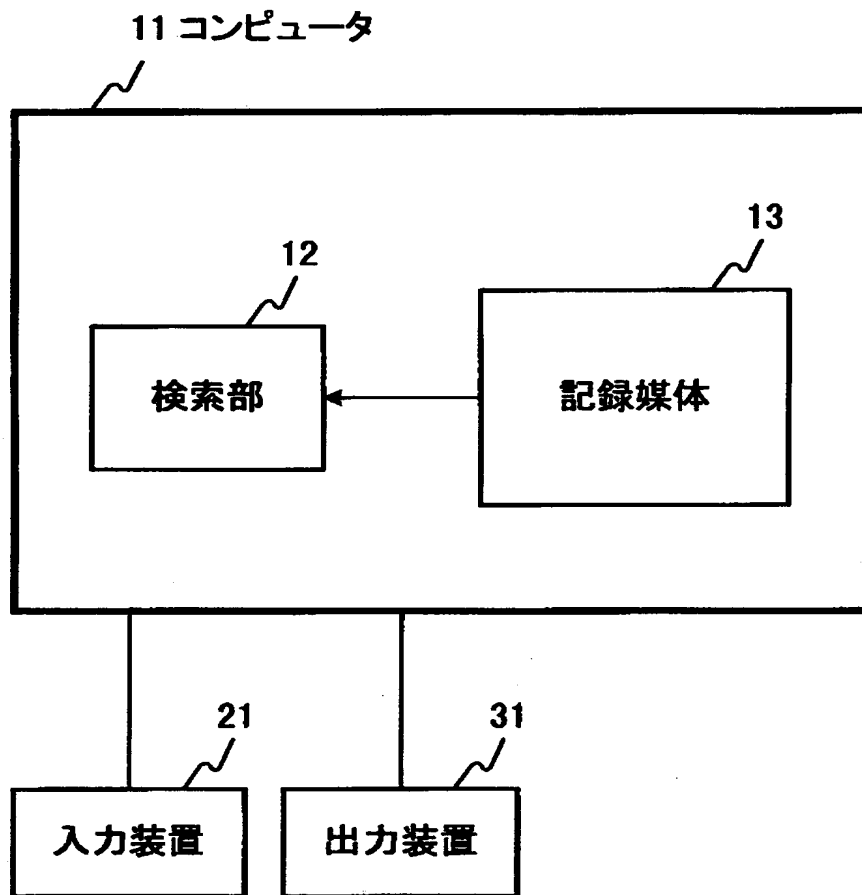
【符号の説明】

11…コンピュータ

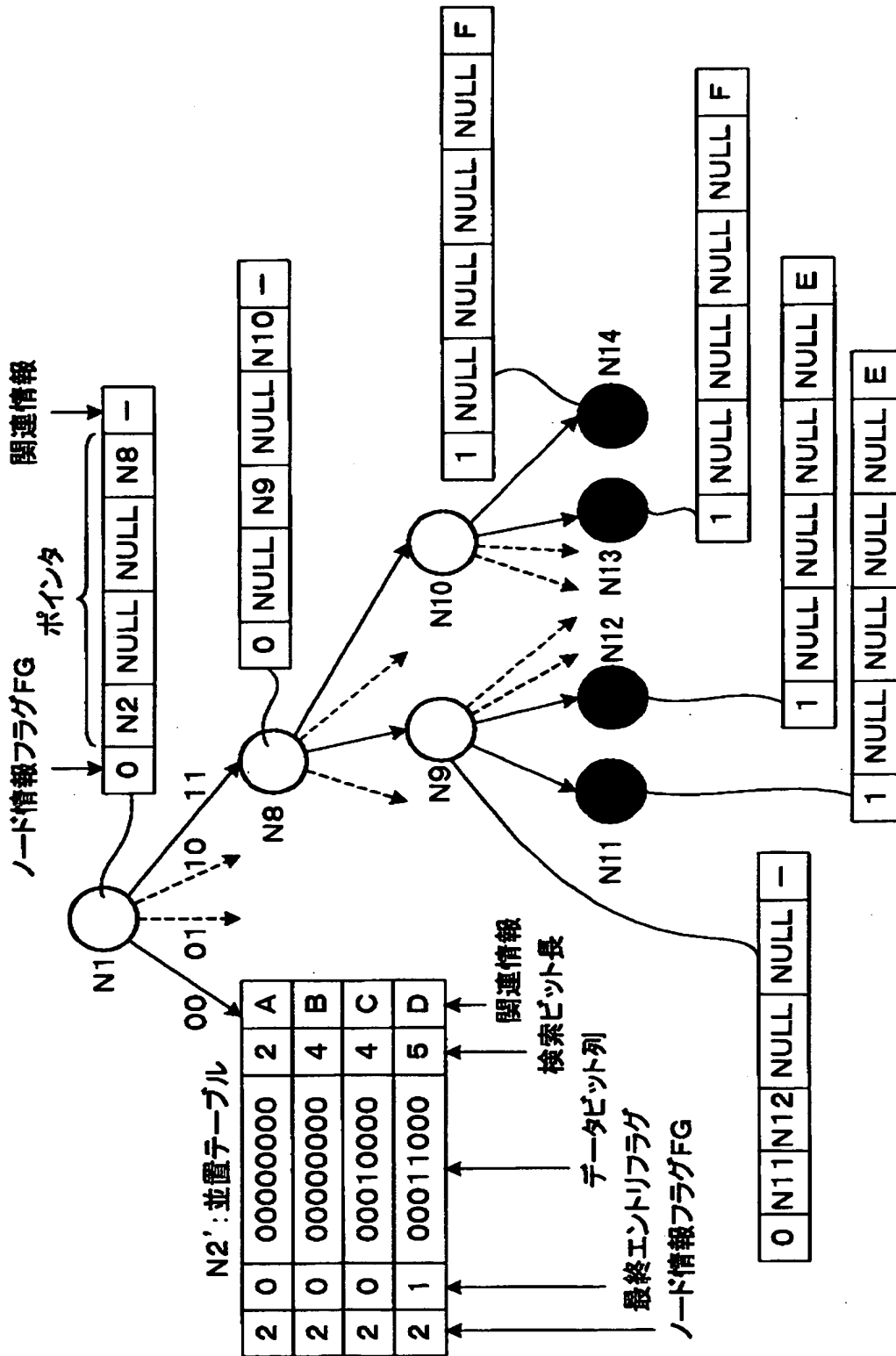
12…検索部
13…記録媒体
21…入力装置
31…出力装置
41…コンピュータ
42…データ構築部
421 …制御部
422 …ツリー構成部
423 …ルートポインタレジスタ
424 …置換ノード選択部
425 …並置テーブル構成部
43…検索部
44…記録媒体
51…入力装置
61…出力装置
71…記録媒体

【書類名】 図面

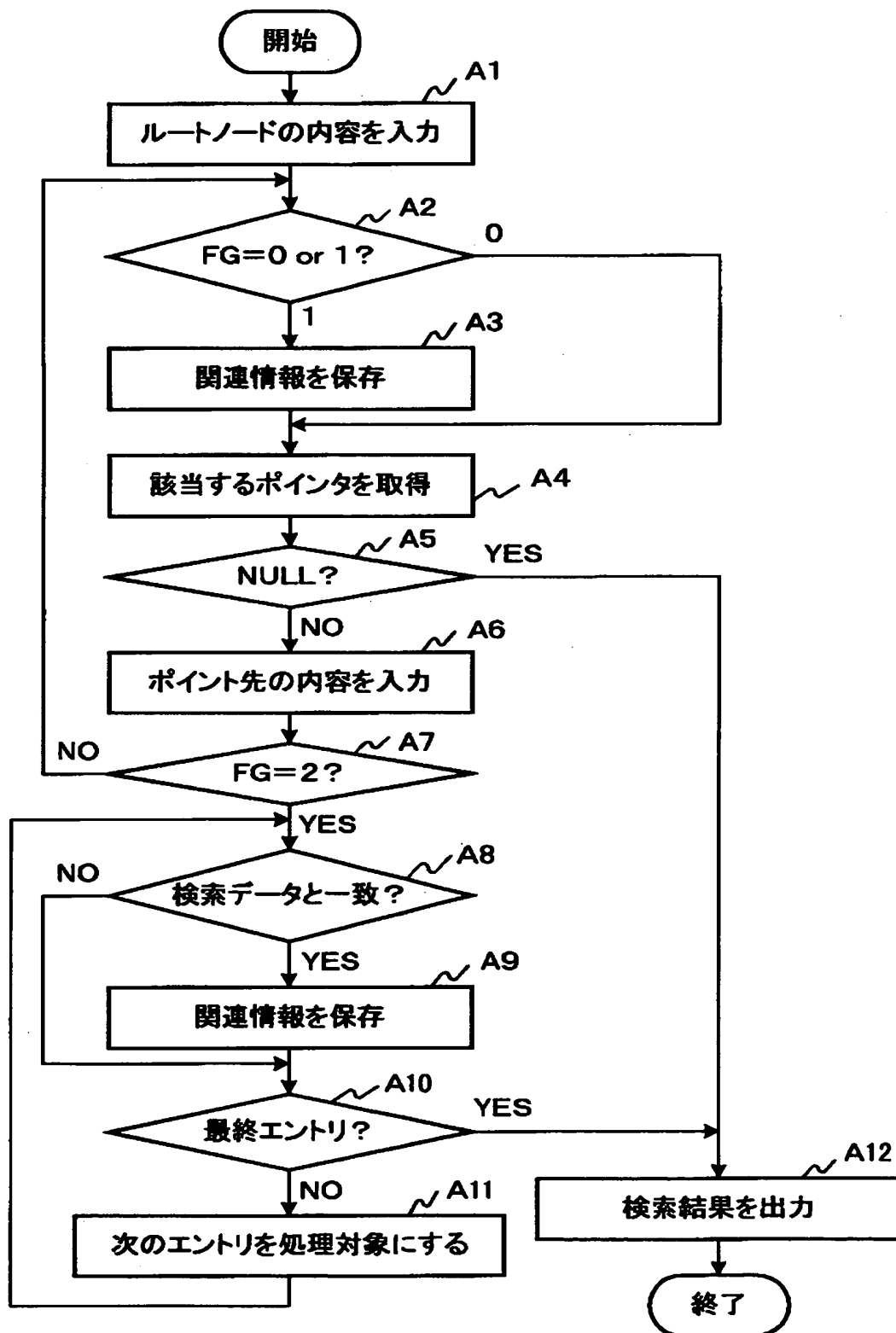
【図 1】



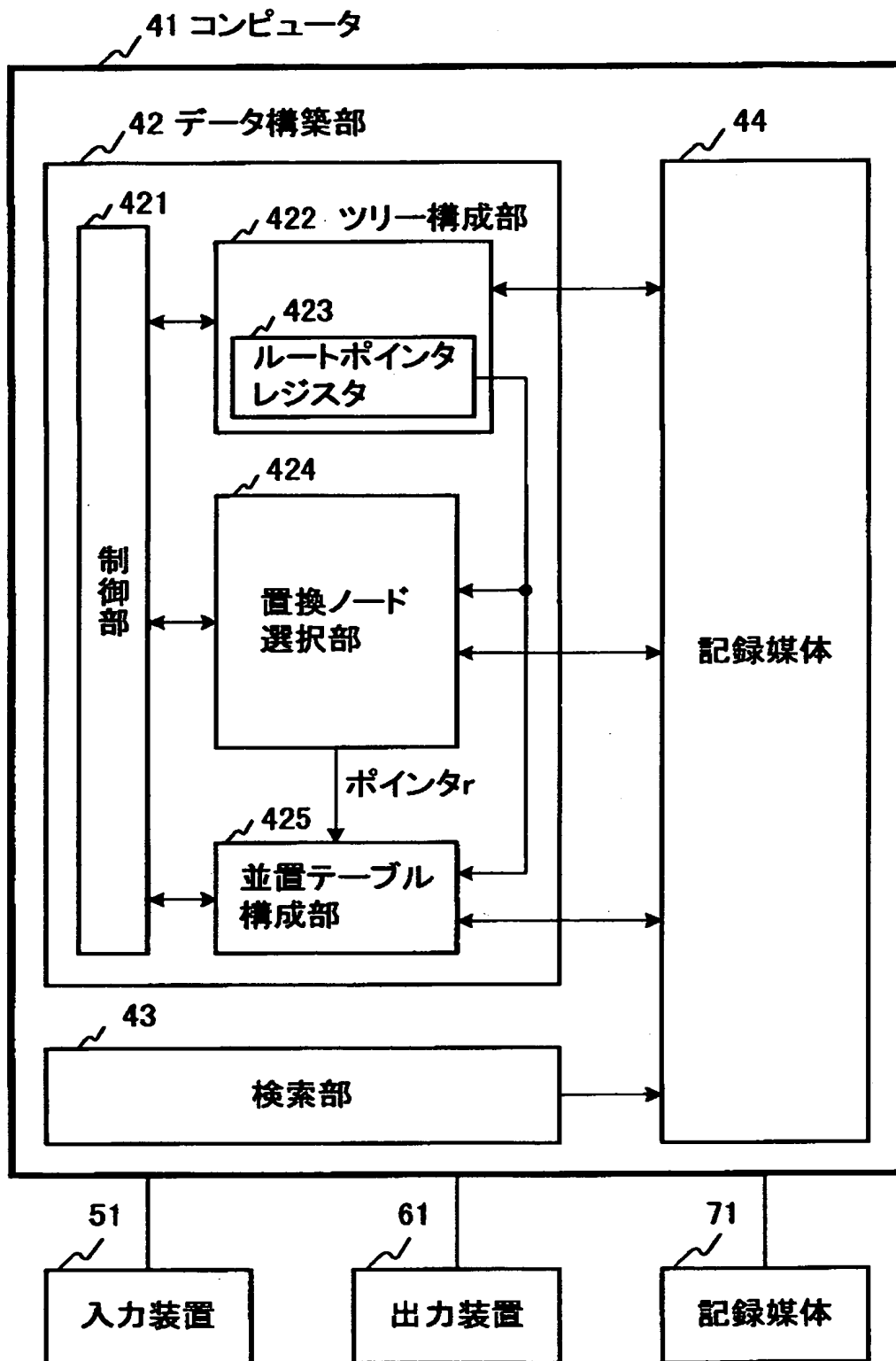
【図2】



【図 3】



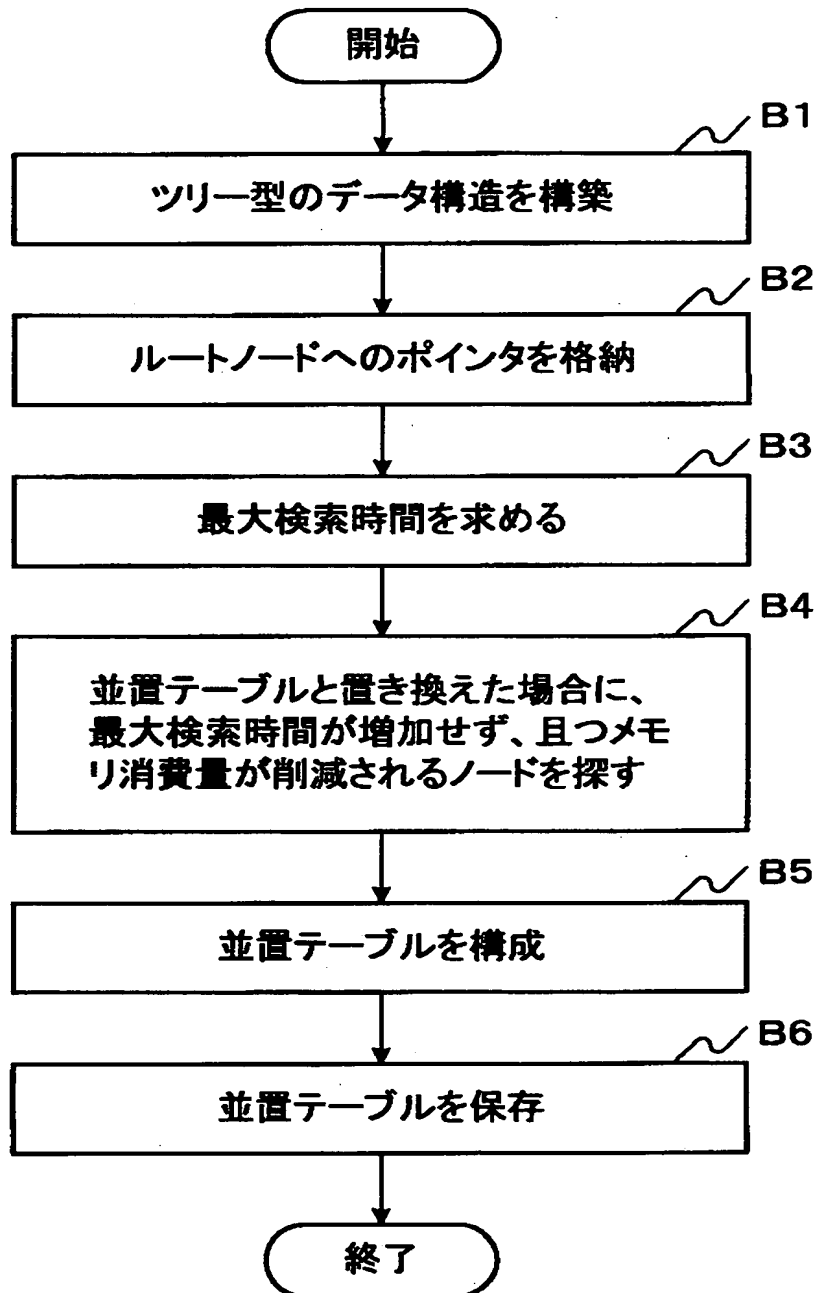
【図4】



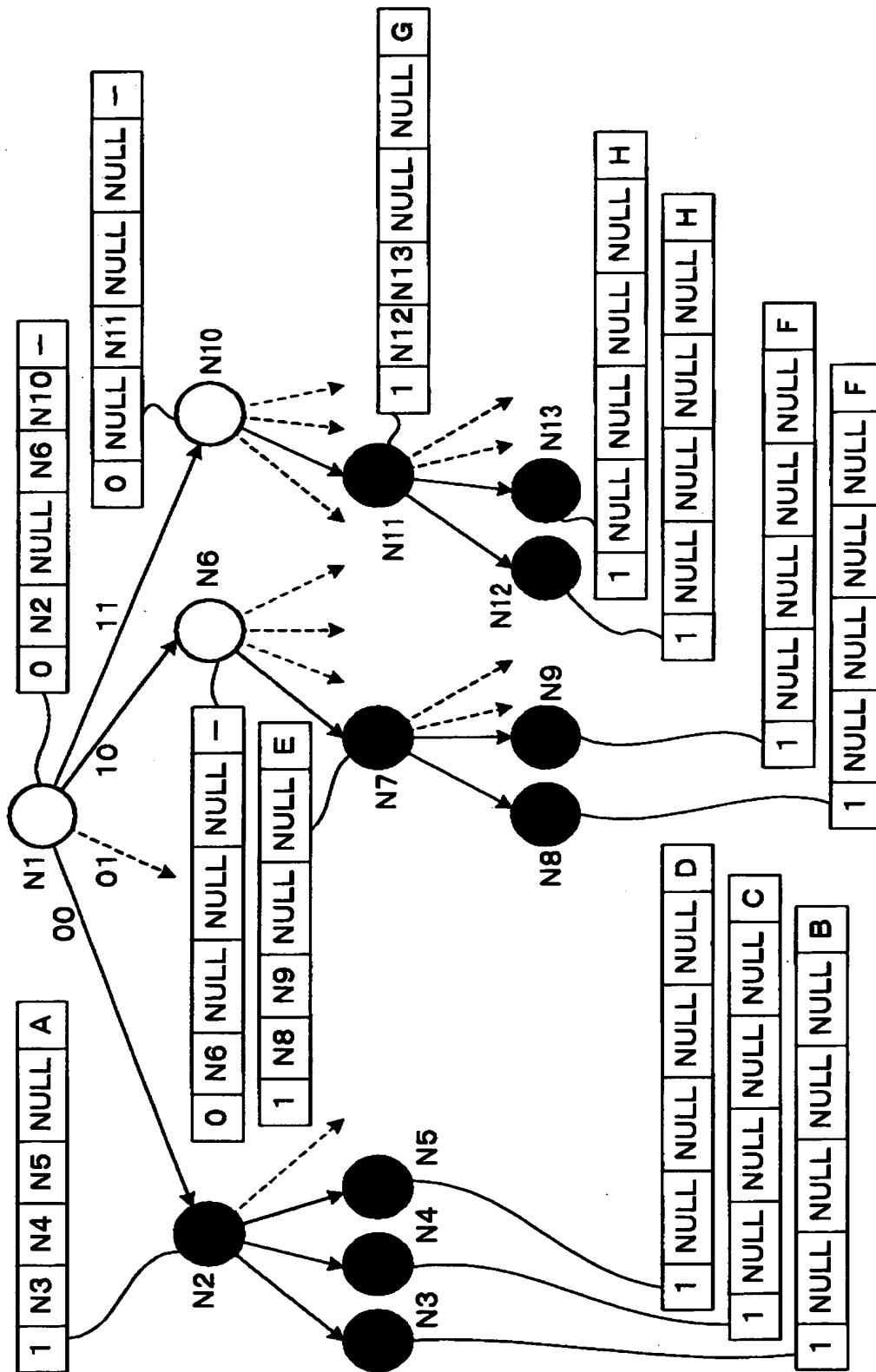
【図 5】

被検索データ	関連情報
00	A
0000	B
0001	C
0010	D
1000	E
10000	F
1101	G
11010	H

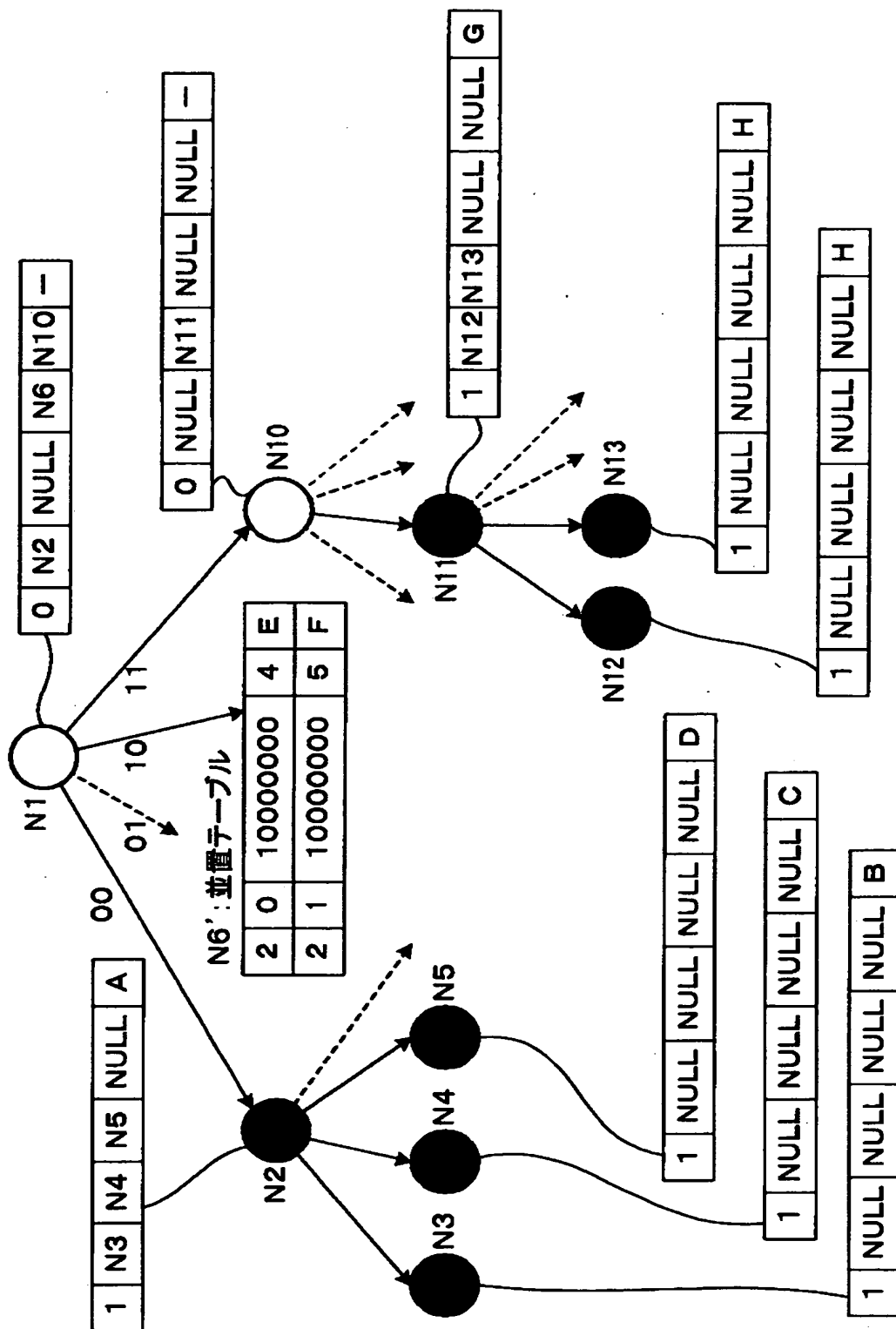
【図 6】



【図 7】



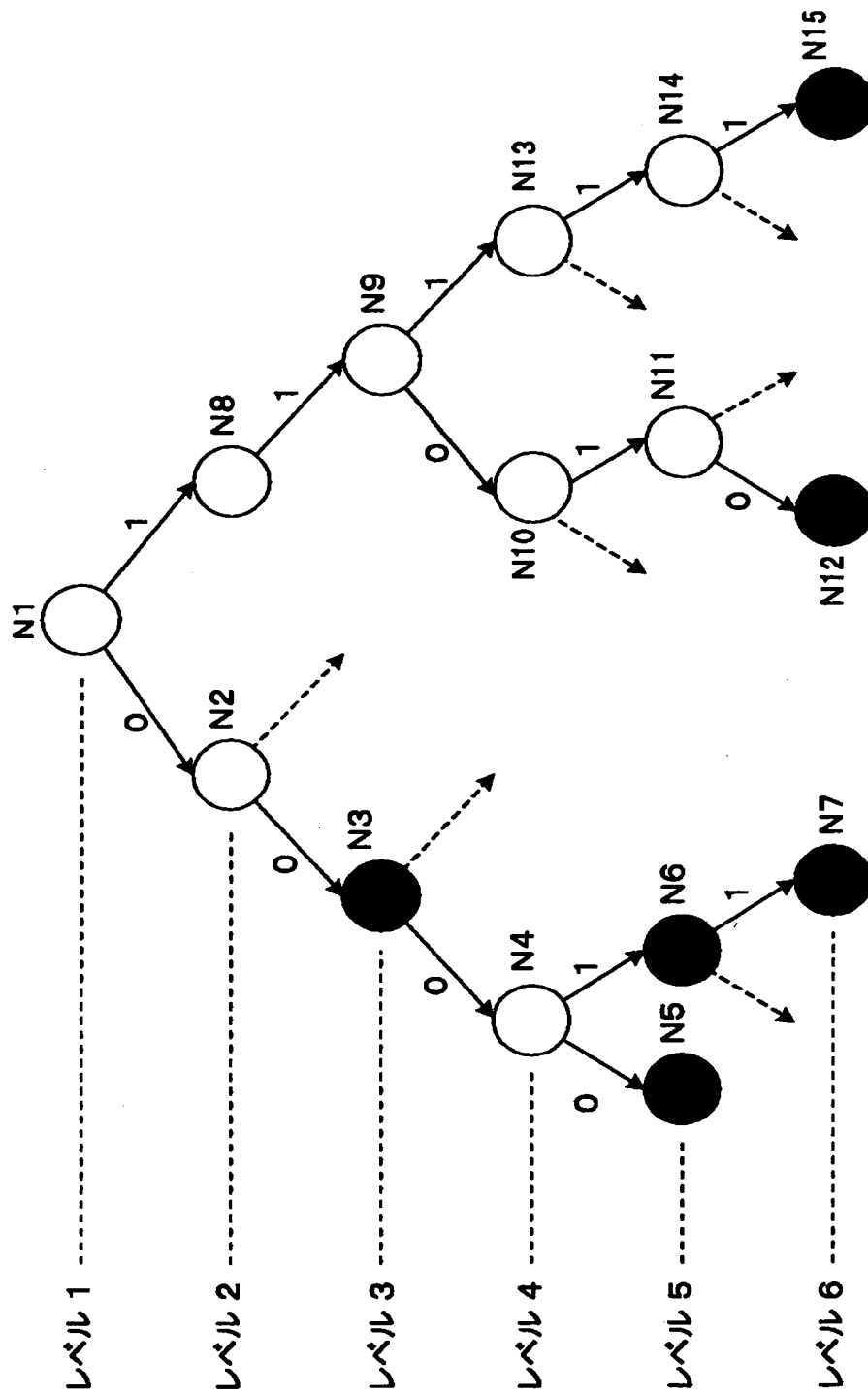
【图 8】



【図9】

被検索データ	関連情報
00	A
0000	B
0001	C
00011	D
11010	E
11111	F

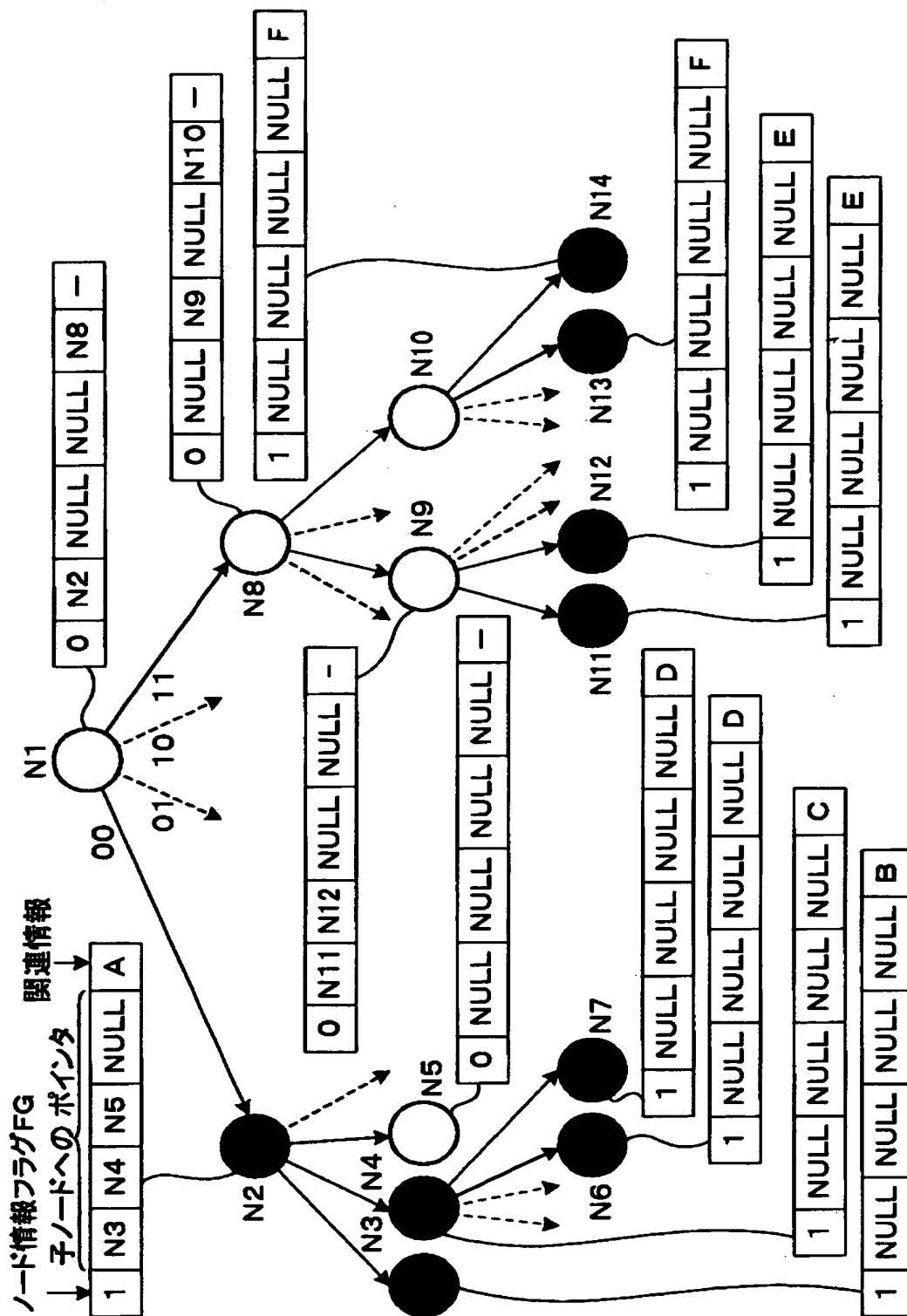
【図10】



【図 1 1】

展開被検索データ	関連情報
00	A
0000	B
0001	C
000110	D
000111	D
110100	E
110101	E
111110	F
111111	F

【図12】



【書類名】 要約書

【要約】

【課題】 データを検索する際の最大検索時間を基数検索法と同等以上にし、且つ、メモリの使用量を基数検索法に比較して少なくする。

【解決手段】 基数検索に用いられるツリー型の検索データ構造の一部に、ツリーの或るノードによって指定されるデータ集合に包含されるような、被検索データの全てを並置したものを、並置テーブルN2' として、上記或るノード以下のすべてのツリー構造の代わりに保持する。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 4 2 3 7]

1. 変更年月日 1 9 9 0 年 8 月 2 9 日

[変更理由] 新規登録

住 所 東京都港区芝五丁目 7 番 1 号

氏 名 日本電気株式会社